

Test Driven Development of Scientific Models

**Thomas L. Clune
NASA Goddard Space Flight Center**

Abstract:

Test-Driven Development (TDD) is a software development process that promises many advantages for developer productivity and has become widely accepted among professional software engineers. As the name suggests, TDD practitioners alternate between writing short automated tests and producing code that passes those tests. Although this overly simplified description will undoubtedly sound prohibitively burdensome to many uninitiated developers, the advent of powerful unit-testing frameworks greatly reduces the effort required to produce and routinely execute suites of tests. By testimony, many developers find TDD to be addicting after only a few days of exposure, and find it unthinkable to return to previous practices. Of course, scientific/technical software differs from other software categories in a number of important respects, but I nonetheless believe that TDD is quite applicable to the development of such software and has the potential to significantly improve programmer productivity and code quality within the scientific community.

After a detailed introduction to TDD, I will present the experience within the Software Systems Support Office (SSSO) in applying the technique to various scientific applications. This discussion will emphasize the various direct and indirect benefits as well as some of the difficulties and limitations of the methodology. I will conclude with a brief description of pFUnit, a unit testing framework I co-developed to support test-driven development of parallel Fortran applications.

**Department of Computer Science and Software Engineering
Auburn University, Auburn, AL, June 8, 2012**